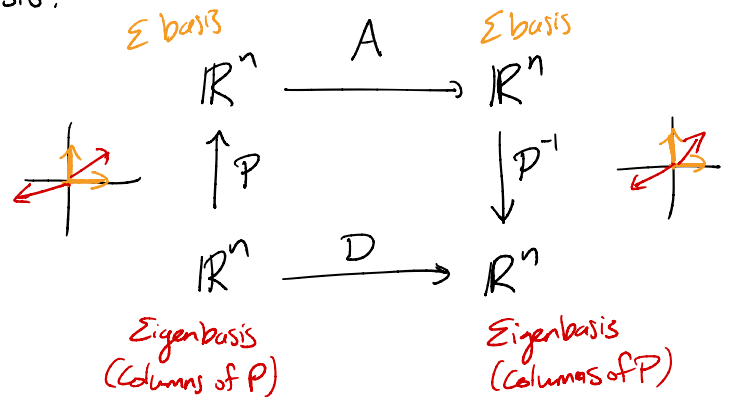
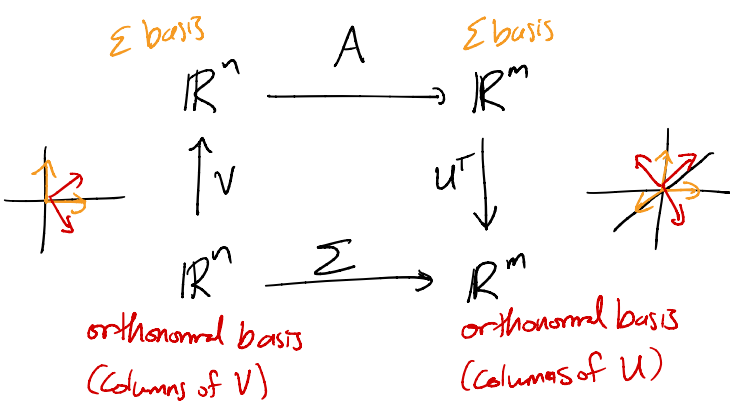
We can view diagonalization and SVD as a change of basis!

$$A = PDP^{-1}$$
$$\Updownarrow$$
$$D = P^{-1}AP$$



Diagonalizing A changes $\mathcal{E}$-basis into a basis of eigenvectors (columns of P). In this new basis, the linear transformation given by A is now given by D, which is much simpler to work with.

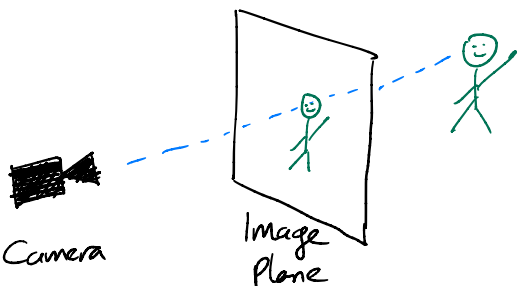
$$A = U\Sigma V^T$$
$$\Updownarrow$$
$$\Sigma = U^T A V$$



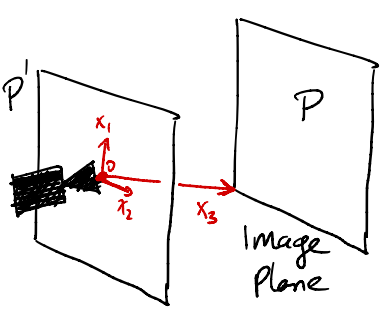
The SVD changes basis of $\mathbb{R}^n$ to columns of V and the basis of $\mathbb{R}^m$ to the columns of U. In these new (orthogonal) bases, the linear transformation given by the L.T. is now given by Σ

Application: Perspective Rendering

In rendering images on a computer, one has to consider perspective. If you want a dog to appear closer than a tree, you have to figure out which pixels make up the dog and which pixels make up the tree. To make this precise, we model what happens when an actual movie is filmed. Filming occurs in $\mathbb{R}^3$, but the images appear in a plane



To easily identify pixels in the image plane P, we use a specific basis for $\mathbb{R}^3$ called the <u>camera basis</u> $\{x_1, x_2, x_3\}$ as shown below. First place the origin of $\mathbb{R}^3$ at the camera. Let P' be the plane parallel to P passing through the origin. Then:

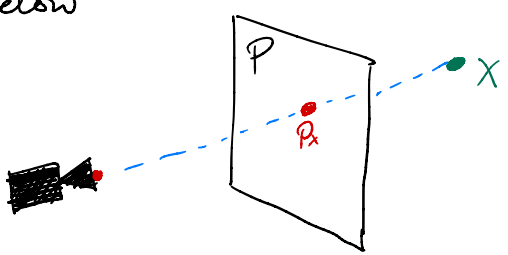


- x_1 points in the horizontal direction of P' and has 1 pixel length
- x_2 points in the vertical direction of P' and has 1 pixel length
- x_3 goes from 0 to the bottom left of P.

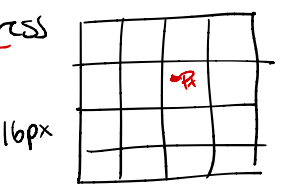
The vector $X = ax_1 + bx_2 + cx_3$ has <u>camera coordinates</u> (a, b, c).

Key Feature: Points in P have camera coordinates of the form $(a, b, 1)$. That is, a vector with camera coordinates (a,b,c) is in P if and only if $c=1$.

Now let x have camera coordinates (a,b,c). We can easily find the corresponding point $p_x \in P$ (assuming it exists) given below

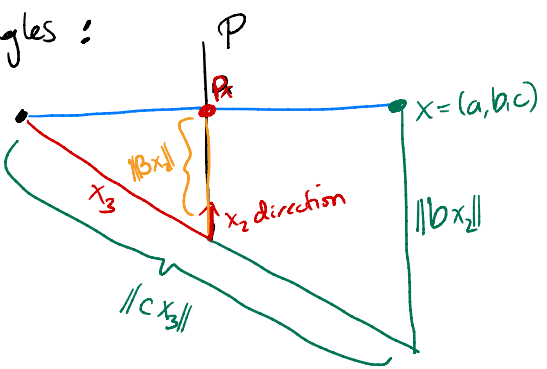


Rounding down the entries of p_x gives the pixel address



Let p_x have camera coordinates $(\alpha, \beta, 1)$.

The side view gives similar triangles:

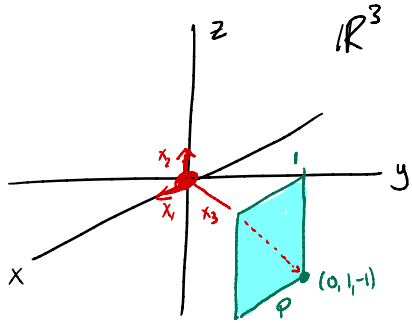


$$\Rightarrow \frac{\|\beta x_2\|}{\|x_3\|} = \frac{\|b x_2\|}{\|c x_3\|} \Rightarrow \beta = \frac{b}{c}$$

Similarly, a top view gives $\alpha = \frac{a}{b}$

Hence if x has camera coordinates (a,b,c), then p_x has camera coordinates $(\frac{a}{c}, \frac{b}{c}, 1)$ and so the pixel address is $(\lfloor \frac{a}{c} \rfloor, \lfloor \frac{b}{c} \rfloor, 1)$ (ie. round down)

Ex: Suppose P is a 1×1 square made of 100×100 pixels centered at $(\frac{1}{2},\frac{1}{2},-\frac{1}{2})$ and the camera is at $(0,0,0)$



Then the camera basis C is:

$$x_1=\begin{bmatrix}1/100\\0\\0\end{bmatrix},\quad x_2=\begin{bmatrix}0\\0\\1/100\end{bmatrix},\quad x_3=\begin{bmatrix}0\\1\\-1\end{bmatrix}$$

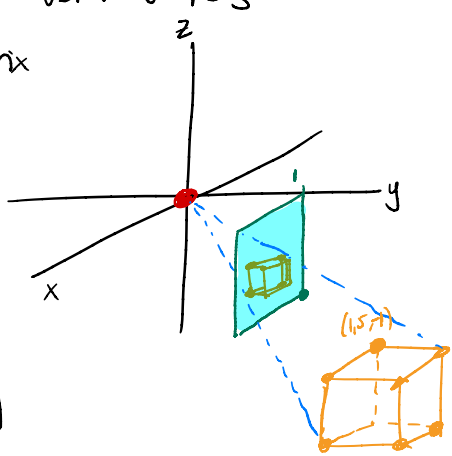
So the change of basis matrix (from $\mathcal{E}$ to C) is $M_C=\begin{bmatrix}1/100&0&0\\0&0&1\\0&1/100&-1\end{bmatrix}$.

Hence the camera coordinates of $x=\begin{bmatrix}a\\b\\c\end{bmatrix}$ are $(100a, 100(b+c), b)$ since $M_C^{-1}\begin{bmatrix}a\\b\\c\end{bmatrix}=\begin{bmatrix}100&0&0\\0&100&100\\0&1&0\end{bmatrix}\begin{bmatrix}a\\b\\c\end{bmatrix}=\begin{bmatrix}100a\\100(b+c)\\b\end{bmatrix}$

and the pixel address of x has camera coordinates $\left(\frac{100a}{b}, \frac{100(b+c)}{b}, 1\right)$ (if it exists)

Now let's render a box whose vertices has $\mathcal{E}$-coordinates arranged in the matrix

$$P=\begin{bmatrix}1&1&3&3&1&1&3&3\\5&7&5&7&5&7&5&7\\-1&-1&-1&-1&-3&-3&-3&-3\end{bmatrix}\begin{matrix}\text{x-coord}\\\text{y-coord}\\\text{z-coord}\end{matrix}$$



First, change coordinates:

$$M_C^{-1}P=\begin{bmatrix}100&100&300&300&100&100&300&300\\400&600&400&600&200&400&200&400\\5&7&5&7&5&7&5&7\end{bmatrix}$$

Now divide each column by its 3rd entry and round down to get the pixel addresses.

$$\begin{bmatrix} 20 & 14 & 60 & 42 & 20 & 14 & 60 & 42 \\ 80 & 85 & 80 & 85 & 40 & 57 & 40 & 57 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

To recover the usual $\mathcal{E}$-basis coordinates, we can multiply by M_C:

$$\begin{bmatrix} .2 & .14 & .6 & .42 & .2 & .14 & .6 & .42 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -.2 & -.15 & -.2 & -.15 & -.6 & -.33 & -.6 & -.33 \end{bmatrix}$$